

The Comparison of Experiments in Embedded System Course

Yong Zhang^{1, a}, Xianghua Hou^{2, b}, Minmin Miao^{1, c} and Honghai Liu^{1, d*}

¹ School of Information Engineering, Huzhou University, Huzhou, China.

² School of Electronic Information, Huzhou College, Huzhou, China.

^a 03086@zjhu.edu.cn, ^b 386172626@qq.com, ^c 02746@zjhu.edu.cn, ^d 204850763@qq.com

Abstract. The embedded system is designed to perform specific computation tasks and usually embedded in other devices. Unlike normal personal computer and cloud server, it is not for daily use but production environment with higher stability and reliability demands in various fields with fixed functionalities, smaller size and lower power consumption for the use of portable and power limit situations. The course of embedded system generally contains fundamental concepts, hardware architecture, software development, bootloader and kernel development, etc. It is a course closely integrating theory and experiments, with the purpose of making resource optimization and management, power efficiency, security and integrity. As a course in university, there are many factors that teacher needs to take into consideration before conducting the experiments in embedded system. In this paper, various factors relevant to the experiments, such as the system architecture, platform, and principles, are discussed in details for the comparison of embedded system experiments.

Keywords: embedded system, hardware architecture, assembly program.

1. Introduction

It has been several decades since the first embedded system came into being. In early years around 1940s, the compact embedded system was invented with the name of Apollo computer for monitoring and controlling spacecraft operations. When the time went to 1970s, the significant revolution on embedded system was the microprocessor, as intel released the world's first microprocessor named 4004, a 4-bit central processing unit, in the year of 1971. During the 1980s, a variety of electronic devices powered by embedded system arose and had great influences on people's daily life. With industrial scales growing up rapidly, there had been great demands for automation systems to perform complex tasks automatically, which also contributed to the development of embedded system in automation engineering equipment. Internet has become popular since the 2000s, when embedded system has been connected to the Internet, enabling people to control systems easily and remotely. To achieve the goal of high portability, mobile phones and tablets were then invented with various sensors for detecting specific aspects, which is mainly based on embedded system as well. Recently, the Internet of Things and artificial intelligence have become the most popular trend for embedded system, guiding it for a more advanced development[1]. The course of embedded system is to bring a solid knowledge on the embedded architecture, operation platform, hardware and software development. It usually requires microcontroller assembly basis and operation system courses as its pre-requisites, and encourages students to develop their own embedded system with core modules and implementations[2]-[4].

2. The Comparison of Embedded System Course

2.1 System Architecture

The training goals of operating system course in Chinese universities are usually to master the relevant concepts of operating systems, understand the implementation principles of key modules and functionalities, and conduct corresponding experiments based on theoretical courses, so that students can understand the core principles of typical operating systems and prepare themselves for learning other software and hardware in the future with a solid foundation of knowledge[5]. An

analysis of the selected operating system textbooks in some Chinese universities shows that there are currently a large number of textbooks that not only explain the basic knowledge of operating systems required by undergraduate students, but also focus on the training of students for postgraduate examinations. The training goals of different types of teaching materials have advantages and disadvantages for students learning operating system courses. Too much emphasis of postgraduate examination during the undergraduate stage has major disadvantages for this course. Students using such teaching materials are prone to deviating from fundamentals. They may be good at participating in various class examinations and assessments, but lack of understanding of the operating system itself, making it difficult to deal with various problems that arise in the actual production environment after entering the workplace.

However, the training goals of foreign universities are on solid operating system understanding, critical thinking and problem-solving skills, professional research and leadership thinking, self-reflection and summary, etc. For college graduates who have completed operating system course, they should have in-depth understanding and insights into the principles of the operating system itself, and be able to apply the knowledge they have learned to practical projects under the guidance of teachers. Critical thinking and question solving skills enable them to efficiently solve various problems, which is conducive to get comfortable at work environment as soon as possible after graduation, while daily self-thinking includes thinking about the past, the present and the future[6]. On the basis of facts, we should actively discover the potential existing problems and take measures in time to avoid the similar problems in future study and work.

2.2 Development Platform.

There are generally two platforms for development and experiments of embedded system, which are Microsoft Windows and GNU/Linux[6] operating systems. The overall selection between the two can be just based on the teacher's preferences, but GNU/Linux is taken in this paper as the experiment platform with the following reasons. The philosophy of Linux is to bring a more flexible, stable, and free environment for designing and building software. With the ultimate spirit of open source and free software, the source code of GNU/Linux is publicly available for everyone, making it possible for students to learn the internal working principles and contributing to the experiment students conduct at class. During the learning and practice process, there is no doubt that various problems might arise and become the barriers for students to conduct further experiments. With the huge members active in Linux open source communities distributed in various fields, students can search and question in communities where many other people are willing to assist, making the learning of this course more effective and close to the production environment[7].

Please note that, during the teaching process, it is common that students are likely to worry if GNU/Linux platform is hard to get started. To solve the problem and get comfortable to the experiment platform, there are some suggested measures that can be taken. For students without experience using GNU/Linux, teacher may start by introducing the history of GNU/Linux, compare it with Microsoft Windows, and bring clear comparison of both operating system on its use and functionalities. As most of students are used to Microsoft Windows for its user interface and basic operations, choosing a distribution that is similar to Microsoft Windows may be a good approach, such as Ubuntu and Linux Mint, whose user interfaces are relatively more friendly and easy to use. Another negative fact is, for students lack of patience, that they can get bored easily without some significant achievement and goal realization. To keep students' initiative and patience to perform further study, specific experiments should be included while they are getting used to the experiment platform.

Other than the operating system, the assembler to assemble low-level programs also makes a difference on students conducting experiments[8]. An assembler takes assembly language code as input and converts it to machine code that can run on the computer directly. Famous assemblers available for x86 are MASM and NASM, and there are obvious differences on the syntax and platform compatibility. MASM was developed by Microsoft corporation, integrated closely with

Microsoft development software and libraries, making it more practical on Windows platform. On the contrary, NASM is a cross-platform assembler, supporting operating systems such as Microsoft Windows, GNU/Linux, MacOS, which meets the requirements of conducting embedded system experiments for students in university. For the syntax difference, compared to MASM syntax, case-sensitive is vital for NASM. NASM assembler treats letters with upper and lower case differently, and to convert all letters to uppercase, the uppercase directive can be invoked. Nowadays MASN syntax is as well supported by NASM assembler with the version newer than 2.15, bringing more flexibilities for people that are used to programming with MASM syntax..

2.3 Experiment Conduction

The experiment class usually takes half of the total teaching class of embedded system course, which consists of various experiments based on the outline of the teaching goals. For many Chinese universities, the experiments are mainly focused on the validation of basic embedded modules. Taking the addressing modes for x86 as example, the mode contains register addressing, immediate addressing, direct memory addressing, direct offset addressing and register indirect addressing. In the textbook, there is detailed introduction of each mode and operation instruction examples, and when it comes to the experiment, students are given specific conditions and requests to write the assembly code for obtaining and manipulating the data from target address, which is basically all the experiment about. Thinking from the situation where the students participate in the course, there is barely no personal development students can make by the embedded system experiment like addressing data, making students bored and lack of power that drives them for deeper research. Without understanding the working principle of system modules, students take much more energy to remember the source code instead of understanding the system principles when it comes to examinations, bringing them more pressure for the course learning and negative impacts on the teaching results.

However, in some foreign universities, the experiments can be more practical and task-driven. The first experiment of the university is the boot sector program, whose preparation is simple and easy to begin with. It is a non-standard embedded operating system that students can create in just 10 minutes. Students don't have to completely understand how it works in the beginning, but are able to make significant achievement within a short time, which is usually perfect to catch students' attention and curiosity, bringing more learning initiative in the future study. After completing the first experiment on the boot sector program, students can see the string displayed on the screen, after which the teacher explains each line of codes in details. Only few assembly instructions which consist of org, mov, call, jmp and so on, are used in the first experiment. The teacher introduces each instruction functionality, syntax and the corresponding examples implemented by the instruction. Such as the mov instruction, with NASM assembly syntax, to move the data from cs register to ax register, the code starts with mov command, with a whitespace separated, there is the target register and then another whitespace followed by the source register. After completing the experiment source code, the teacher then demonstrates how to assembly the source code to the machine language by invoking NASM assembler. Upon target program file is generated, the teacher then writes the binary data into the target boot device, which can be either a floppy drive or hard drive. The last step to finish this experiment is to make the embedded system boot from target device, till the system boots up and students see the prompt strings, the experiment is complete.

2.4 Basic Principles

As many Chinese students are used to absorbing the knowledge from the teachers since primary school, they may be good at learning theories from text book, but some obstacles might come when it turns to experiment. Teacher of embedded system course should be prepared to help students overcome the difficulties in advance before students get frustrated at this course.

To make students efficient and happy to learn[9]. For people graduated from information engineering major, the first hello world program is always hard to forget for decades. The sense of

accomplishment comes from the several lines of code, the process to debug program, and gaining of the expected result. But many Chinese textbooks of embedded system contains too many abstract descriptions of system working principles such as process dispatching algorithm, without a complete result that can motivate students to learn.

To learn from sensibility to rationality. Before people see the ocean in reality, no matter how vivid the author describes in book with pictures and experience, people still find it far away from themselves and hard to get the feelings of the ocean. Once people have already explored the ocean themselves, there is no problem that they can describe the scene with their own thoughts much more vivid than the text in the book. Similar theory applies to the embedded system experiment as well, which means that the teacher is the guide for students to explorer the ocean of embedded system. Common textbooks are just designed to describe what the ocean looks like and how it feels with the ocean, but the experiment of developing students' own embedded system is to feel the ocean, getting themselves involved and swimming freely in the ocean of embedded system.

To learn the most needed knowledge in limited time. Teachers and students believe that learning embedded system course is not just for the final examination but for the personal development and hobbies. Various foreign researches indicate the average time for students staying focused is 15 minutes in each class, as there is a limitation of people accepting and absorbing new knowledge in a certain period of time. To make the embedded system course learning more efficient, teacher is encouraged to make full use of the 15 minutes in a class, which limits the overall quantity of knowledge the teacher introduces as well. The fun fact is, students are actually able to learn more things in a short time, with solid impression lasting for a long time. Meanwhile, teacher may try and break the experiment learning into several parts in a class, with little break or discussion in between, which probably helps students reduce the mental and physical tiredness.

To stay confident and learn gradually. Confidence makes a great difference on performing embedded system experiments. The course consists of a wide range of theories, which makes the experiments difficult than many other courses. Various warnings and errors might occur while performing experiments, which requires the confidence to solve. Being confident also contributes to the persistence of learning and energetic attitude, enhancing the skills of trying different approaches to realize the experiment goals. As teamwork is common during the experiment class, confident students are more likely to lead the team and benefit more from the team discussion. While participating the teamwork, students do not have to understand everything the first time they try, and the teacher may encourage them to understand the brief knowledge, and learn gradually and repetitively. Taking the above boot sector program as example, there is no worry if students find it hard to figure out the involved assembly instructions after class, but with experiments going on, once students look back, they will probably have a better understanding of assembly instructions than the first time they try during the experiment class.

3. Experiment Assessments

Assessments of embedded system experiments can vary a lot among different universities, but it mainly depends on the experiment report, source code programming and daily assignments[10]. In most cases, students are asked to complete the experiment report after each class of embedded system, and the experiment report takes the most weight of the overall course scores. Students are advised to write experiment goals and content, modular source code and operation procedures. To check if the result meets the experiment goals, the screenshot of running program is inserted into the experiment result chapter. At the end of the report, there are self-reflection and experiment summary as well, where students list the achievements they gain and the problems encountered during performing the experiment and solutions towards them.

For example, in the boot sector program, the report goals may include the basic assembly instructions and usage, NASM assembler operations, and Linux commands to copy assembled program to the boot device. In the source code chapter, all the assembly code is asked to put into the

report, with code comments followed after each line of code, explaining the purpose of each code. For the operation procedures, there are mainly five steps, which are listed the following. First step is to enter the destination directory by `cd` command in the terminal, and then students need to create the text file named `exp01.s`, where the `s` extension name indicates assembly source code. After the text file is saved, it comes to the third step, and the source code file should be assembled to binary file by using `nasm` command and passing the source and target file names. The fourth step is to copy the binary data from the assembled file to the target device, and then make the embedded system boot from the target device. If everything goes well, the embedded system boots with the hello world string printed on the screen, which is the result of operating graphical memory instead of invoking normal `printf` function[11]. In the summary of the report, students may write the mistakes they made during the experiment and raise some questions of source code if necessary as well. To evaluate such experiment report, teacher may focus on the basic format alignment, there are obvious misalignment of text in some reports, which may indicate a report that is merged from various sources. Usually there is no problem on the source code and corresponding comments, but the screenshot of the program result. Some students may not conduct the experiment personally but copy the result picture from some other students' report. By inspecting the path of the desktop directory, in many cases other than administrator or some username like this, the username inside the path route should not be the same. The experiment score is reduced when cheating behaviors are caught and students should be informed to take the consequence of cheating in experiment assignments.

4. Summary

The history of embedded system is introduced in the beginning according to the time order, from early spacecraft control to artificial intelligence, which became an important course for students in computer major nowadays. The embedded system major for university students contains both theory and experiment learning, while the experiment takes more weight in the final score of results. The architecture of this system might vary from teacher's preference, and in this paper, the x86 processor is chosen for introduction. The experiment platforms can be complex as it depends on both the hardware and software environments, while for the software, the base operating system and assembler make a difference on developing the target embedded system. While conducting the experiments, some Chinese universities only focus on the theory validation instead of the actual functionality, making students bored and lack of learning initiative. Other than this problem, there are some other principles that teacher needs to follow to make students learn efficiently with sensibility, and not to demand students to understand everything for the first time, but to learn the embedded system experiments gradually with stronger persistence, which brings a better teaching effect for this course. Assessment of embedded system experiment require multiple aspects and for Chinese students the experiment report is the most fundamental material, but some students may not complete the report by themselves. There is still a long way to go for teachers to bring students more learning initiative of the embedded system course for conducting the corresponding experiments.

Acknowledgment

This work was financially supported by Huzhou University Research and Design Fund under Grant 2021XJKJ08, the school-enterprise cooperative project under the Ministry of Education under Grant 201701060011, the Huzhou College Teaching Reform Project JY64020 ,titled “ Research on Teaching Reform of Operating System Couese from the Perspective of Course Ideology and Politics”, and the Zhejiang Province's "14th Five Year Plan" Graduate Teaching Reform Project, titled “ Exploration and practice of innovative ability training model for Master's degree students majoring in electronic information -- Taking Huzhou University as an example.

References

- [1] Qin Guorui, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Teaching reform and research on the "Operating System Principles" course in undergraduate colleges and universities. Computer Knowledge and Technology, 2021.
- [2] Pei Lichen, Zhao Xuezhuan, Construction of a multi-level task-driven operating system principle teaching system. Computer Knowledge and Technology, 2019(35).
- [3] Liu Hailing, Ge Jike, Zhu Huazheng, Strategy and practice of ideological and political education in the "Principles of Operating Systems" course. Heilongjiang Education (Theory and Practice), 2022(02).
- [4] Wang Junchao, Yao Jinyang, Shan Zheng, Liu Fudong, Hou Yifan, "Three-dimensional" teaching reform of the operating system principle and analysis course for cyberspace security majors. Computer Education, 2020(10).
- [5] Zhang Yang, Sun Tengfei, Discussion on the "information heuristic teaching" method of information-based intelligent teaching reform courses. China Industry and Informatization, 2023.
- [6] Shi Yingjuan, Jin Xiaokang, Ren Mingyuan, Yu Li, Shen Caiyou, Jin Hu, Exploration of problem-driven teaching model focusing on the cultivation of computational thinking. Computer Education, 2022(11).
- [7] Dong Yueyu, Qiang Zhenping, Hu Kunrong, Wang Xiaorui, Discussion on the construction of computer system ability training teaching system under the new engineering concept. University Education, 2021(06).
- [8] Zhang Yuzhuo, Xiao Fei, Xiang Yu, Exploration on the teaching practice of "Ideological and Political Curriculum" in computer operating systems. Scientific Consulting (Technology·Management), 2020(08).
- [9] Li Wenjuan, Wang Lidong, Shou Zhouxiang, Zhao Haili, Multi-level practice drives the teaching reform of "Operating System Principles". Computer Knowledge and Technology, 2018(05).
- [10] Ma Honglin, Yan Lei, Yu Junwei, Exploration and practice of "learning-centered" reform of operating system teaching paradigm. Computer Education, 2020(06).
- [11] Zhou Lihua, Zhou Junhua, Xiao Qing, Exploration and practice of ideological and political teaching in the course of computer composition principles. Computer Education, 2022(02).